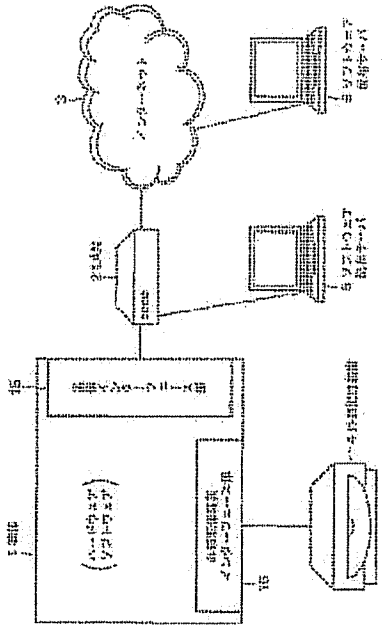
SOFTWARE PROCESSING DEVICE AND SOFTWARE INSTALLATION METHOD

Publication number: JP2003122588 (A)
Publication date: 2003-04-25
Inventor(s): FRANCESCO STAJANO; ISOZAKI HIROSHI +
Applicant(s): TOSHIBA CORP +
Classification:
- international: ***G06F1/00; G06F11/00; G06F12/14; G06F21/22; G06F21/24; G06F9/445;*** **G06F1/00; G06F11/00; G06F12/14; G06F21/00; G06F21/22; G06F9/445;** (IPC1-7): G06F1/00; G06F11/00; G06F12/14; G06F9/445
- European:
Application number: JP20010315815 20011012
Priority number(s): JP20010315815 20011012

Also published as:
JP3863401 (B2)

Abstract of **JP 2003122588 (A)**

PROBLEM TO BE SOLVED: To provide a software processing device capable of obstructing the installation of invalid software. SOLUTION: Downloaded software and an attached signature enciphered by the secret key of a software distributor are temporarily held in the buffer of a RAM 13. After a processor 14 and the working memory of the RAM 13 are initialized at a specified timing and, based on the public key of the distributor created in a ROM 11 and the signature of the software temporarily held in the buffer, the validity of the software is verified by a first processing program code created in the ROM 11. When the validity is verified, the software from the buffer of the RAM 13 is stored in a flash memory 12 by a second processing program code created in the ROM 11.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2003－122588
（P2003－122588A）
(43)公開日　平成15年４月25日(2003.4.25)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　11/00		Ｃ０６Ｆ　12/14	３１０Ｚ	５Ｂ０１７
1/00		9/06	６３０Ａ	５Ｂ０７６
9/445			６１０Ｌ	
12/14	３１０		６４０Ａ	
			６６０Ｇ	

審査請求　未請求　請求項の数20　ＯＬ　（全 12 頁）

(21)出願番号　特願2001－315815(P2001－315815)

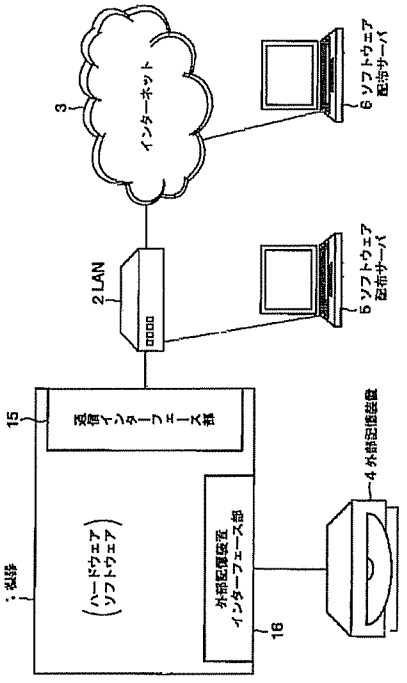
(22)出願日　平成13年10月12日(2001.10.12)

(71)出願人　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(72)発明者　フランチェスコ・スタヤノ
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
(72)発明者　磯崎　宏
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
(74)代理人　100058479
弁理士　鈴江　武彦　（外６名）

最終頁に続く

(54)【発明の名称】　ソフトウェア処理装置及びソフトウェア・インストール方法

(57)【要約】
【課題】　正当でないソフトウェアのインストールを阻止することのできるソフトウェア処理装置を提供すること。
【解決手段】　ダウンロードしたソフトウェア及びこれに添付されているソフトウェア配布元の秘密鍵で暗号化されたシグネチャを、ＲＡＭ１３のバッファに一時的に保持しておく。所定のタイミングで、プロセッサ１４とＲＡＭ１３のワーキングメモリを初期化した後に、ＲＯＭ１１に作り込まれた該配布元の公開鍵と、該バッファに一時的に保持されているソフトウェアに対するシグネチャとに基づいて、ＲＯＭ１１に作り込まれた第１の処理のプログラムコードによって、該ソフトウェアの正当性を検証する。正当性が確認された場合には、ＲＯＭ１１に作り込まれた第２の処理のプログラムコードによって、該ソフトウェアをＲＡＭ１３のバッファからフラッシュメモリ１２に保存する。



【特許請求の範囲】
【請求項１】外部からソフトウェアをインストールして実行する機能を有するソフトウェア処理装置において、
ソフトウェアを実行するためのプロセッサと、
実行すべきソフトウェアを保存するためのフラッシュメモリと、
外部から入力された、ソフトウェアと、該ソフトウェアの正当性の検証のための、第１の鍵で暗号化されたシグネチャとを、一時的に保持するためのバッファと、
前記第１の鍵に一意に対応する第２の鍵と、前記バッファに一時的に保持されている前記ソフトウェアに対するシグネチャと該第２の鍵とに基づいて該ソフトウェアの正当性を検証するための第１の処理のプログラムコード、該検証によって正当性が確認された前記ソフトウェアを前記バッファから前記フラッシュメモリに保存するための第２の処理のプログラムコードとを書き込んだＲＯＭとを備えたことを特徴とするソフトウェア処理装置。
【請求項２】前記フラッシュメモリは、前記ＲＯＭ内の前記第２の処理のプログラムを実行することによってのみ、データ書き込み可能であることを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項３】前記第２の処理は、前記バッファに一時的に保存されている前記ソフトウェアを前記フラッシュメモリに保存されているソフトウェアに対して上書きするのに先だって、前記バッファに一時的に保存されている前記ソフトウェアと、前記フラッシュメモリに保存されている前記ソフトウェアとの新旧を比較し、前記バッファに一時的に保存されている前記ソフトウェアの方が新しいと判断された場合にのみ、前記上書きを行うことを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項４】前記第２の処理は、前記ソフトウェアに付加されている日時情報に基づいて新旧の比較を行うことを特徴とする請求項３に記載のソフトウェア処理装置。
【請求項５】前記第２の処理は、前記ソフトウェアに付加されているバージョン情報に基づいて新旧の比較を行うことを特徴とする請求項３に記載のソフトウェア処理装置。
【請求項６】前記第１の鍵は、前記ソフトウェアの配布元に固有の秘密鍵であり、
前記第２の鍵は、前記ソフトウェアの配布元に固有の公開鍵であり、
前記第１の処理は、前記バッファに一時的に保持されている前記秘密鍵で暗号化されたシグネチャを、前記ＲＯＭ内に書き込まれている前記公開鍵で復号する処理と、該復号によって得たシグネチャに基づいて改竄の有無を検証する処理とを含むことを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項７】前記ＲＯＭを、対象とするソフトウェア配布元ごとに設けたことを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項８】前記フラッシュメモリに保存される前記ソフトウェアは、オペレーティングシステム、アプリケーション及びそれらが必要とするデータからなる現在インストールされているソフトウェアであることを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項９】前記フラッシュメモリに保存される前記ソフトウェアは、オペレーティングシステム、該オペレーティングシステム上で実行されるミドルウェア、アプリケーション及びそれらが必要とするデータからなる現在インストールされているソフトウェアであることを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項１０】ソフトウェアの稼動時に利用される再書き込み可能なワーキングメモリを更に備えたことを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項１１】一定期間ごとに前記プロセッサ及び前記ワーキングメモリの初期化を行うためのリセット手段を更に備えたことを特徴とする請求項１０に記載のソフトウェア処理装置。
【請求項１２】一定期間ごとに、前記プロセッサ及び前記ワーキングメモリの初期化を行った後に、前記ＲＯＭに書き込まれた前記処理に制御を移すタイマー手段を更に備えたことを特徴とする請求項１０に記載のソフトウェア処理装置。
【請求項１３】前記タイマー手段は、オペレーティングシステム、ミドルウェア及びアプリケーションのいずれの処理とも独立し、それら処理からは制御不可能で、前記一定期間の経過を管理することを特徴とする請求項１２に記載のソフトウェア処理装置。
【請求項１４】前記タイマー手段は、前記ＲＯＭに蓄えられた値を元に一定周期で値を加算することによって、前記一定期間の経過を管理することを特徴とする請求項１３に記載のソフトウェア処理装置。
【請求項１５】前記タイマー手段は、現在実行中のソフトウェアが次の処理単位を実行する前に、次のリセット予定時間までの残り時間を調べ、残り時間が短かった場合に、通常の処理手順は短時間でリセット可能な状態になるよう処理を行い、該次のリセット予定時間より前にリセットを繰り上げて実行するための手段を含むことを特徴とする請求項１２ないし１４のいずれか１項に記載のソフトウェア処理装置。
【請求項１６】インストールするソフトウェアを取得するための通信インターフェース手段を更に備えたことを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項１７】インストールするソフトウェアを取得するために、ソフトウェアを含むメディアを接続するための外部記憶装置接続インターフェース手段を更に備えたことを特徴とする請求項１に記載のソフトウェア処理装置。
【請求項１８】前記ソフトウェア処理装置は、外部から

ソフトウェアをインストールして実行する機能を有する、家電機器、汎用計算機、携帯電話又はＰＤＡであることを特徴とする請求項１に記載のソフトウェア処理装置。

【請求項１９】外部からソフトウェアをインストールして実行する機能を有する装置におけるソフトウェア・インストール方法であって、外部から入力された、ソフトウェアと、該ソフトウェアの正当性の検証のための、第１の鍵で暗号化されたシグネチャとを、バッファに一時的に保持するステップと、ＲＯＭに書き込まれた前記第１の鍵に一意に対応する第２の鍵と、前記バッファに一時的に保持されている前記ソフトウェアに対するシグネチャとに基づいて、前記ＲＯＭに書き込まれた第１の処理のプログラムコードによって、該ソフトウェアの正当性を検証するステップと、前記検証によって正当性が確認された場合に、前記ＲＯＭに書き込まれた第２の処理のプログラムコードによって、正当性が確認された前記ソフトウェアを前記バッファからフラッシュメモリに保存するステップとを有することを特徴とするソフトウェア・インストール方法。

【請求項２０】一定期間ごとに、プロセッサ及びワーキングメモリの初期化を行った後に、前記ＲＯＭに書き込まれた前記処理に制御を移すステップを更に有することを特徴とする請求項１９に記載のソフトウェア・インストール方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、外部からソフトウェアをインストールして実行する機能を有する、家電機器、汎用計算機、携帯電話、ＰＤＡなどのソフトウェア処理装置及びソフトウェア・インストール方法に関する。

【０００２】

【従来の技術】近年、インターネットの普及や常時接続の一般化に伴い、汎用の計算機をはじめとする多くの機器がネットワークに接続し、利用者はネットワークを利用した様々なサービスを受けることが可能となってきている。その一方で、悪意のある利用者によるネットワークを経由した不正行為が社会的な問題になっており、ネットワークセキュリティの重要性は高まってきている。

【０００３】一般的にそれらの不正行為は、ソフトウェアの脆弱性や管理者の設定ミスを利用した攻撃やウィルスによるものが多い。特にソフトウェアの脆弱性に関しては、開発者にとってあらかじめ設計上、実装上のミスのない完全なソフトウェアを作成し配布することは一般的に困難であることが原因となっている。さらに、それらのミスはソフトウェアの配布後、開発者や利用者によって使用中に発見されることが多いため、開発者側から修正プログラムという形で公開・配布され、利用者は適宜それらの修正プログラムをダウンロード後、インストールするという形態が一般的である。

【０００４】しかしながら、全てのネットワーク利用者がネットワークセキュリティに関して専門の知識や関心を持っているとは限らない。そのため、それらの問題のあるソフトウェアを修正するプログラムが公開されているにも関わらず、問題を抱えたソフトウェアが放置されている場合が少なくない。従って、ソフトウェア作成者が修正プログラムや、修正プログラムを含んだアップデートプログラムを、機器利用者に対して配布し、確実に修正プログラムやソフトウェアをインストールすることが求められる（ここで、確実とは、ダウンロードしたプログラム自体が改竄されていたり、不正なソフトウェアに置き換えられたりすることなく、正当なソフトウェアをインストールすることを指す）。

【０００５】従来のソフトウェアを安全にインストールする技術としては、デジタル署名が有用であると考えられる。デジタル署名の利用方法例を以下に示す。

【０００６】ソフトウェア作成者は、公開鍵アルゴリズムに基づいた公開鍵と秘密鍵を作成し、インストールの対象となる機器のＲＯＭに公開鍵を書き込む。このため、公開鍵を書き換えることは不可能である。一方でインストールするソフトウェアはそれ自身が正しいと証明するため、前記のＲＯＭに書き込まれた公開鍵に対応する秘密鍵で暗号化されたシグネチャを持つ。ソフトウェアをインストールする際、前記の公開鍵とシグネチャを用いて、そのソフトウェアが鍵の作成者、すなわち正当なソフトウェアの作成者であるかソフトウェアの妥当性を検証する。改竄検証のコードもＲＯＭに保存しておけば、例えソフトウェアが悪意のあるコードによって改竄されたとしても、改竄検証の処理が改竄されることはないため、不正なシグネチャを持つソフトウェアのインストールを防ぐことができる。

【０００７】しかしながら、インストール処理を行うソフトウェア自身が実装上、設計上のミスを利用した悪意あるコードによって、改竄検証処理をスキップさせたり、改竄検証結果を無効にさせたりするように改竄されてしまった場合、強制的に正当な改竄検証処理を行わせることはできない。従って前記の手法は、安全にソフトウェアをインストールするための本質的な問題解決にはなっていない。

【０００８】ソフトウェアを安全にインストールさせるためには、ソフトウェアのインストールの際に、必ずシグネチャの改竄検証処理を行うよう、ソフトウェアの改竄を不可能にするだけでなく、ソフトウェアをインストールさせる処理そのものを改竄されずに、安全にソフトウェアのアップデートが行えるようにする仕組みを備えていなければならない。

【０００９】

【発明が解決しようとする課題】以上説明したように、従来のソフトウェアを安全にインストールさせる仕組み

では、正当でないソフトウェアがインストールされる可能性があり、対策として不十分であった。

【００１０】本発明は、上記事情を考慮してなされたもので、正当でないソフトウェアのインストールを阻止することのできるソフトウェア処理装置及びソフトウェア・インストール方法を提供することを目的とする。

【００１１】

【課題を解決するための手段】本発明は、外部からソフトウェアをインストールして実行する機能を有するソフトウェア処理装置において、ソフトウェアを実行するためのプロセッサと、実行すべきソフトウェアを保存するためのフラッシュメモリと、外部から入力された、ソフトウェアと、該ソフトウェアの正当性の検証のための、第１の鍵で暗号化されたシグネチャとを、一時的に保持するためのバッファと、前記第１の鍵に一意に対応する第２の鍵と、前記バッファに一時的に保持されている前記ソフトウェアに対するシグネチャと該第２の鍵とに基づいて該ソフトウェアの正当性を検証するための第１の処理のプログラムコード、該検証によって正当性が確認された前記ソフトウェアを前記バッファから前記フラッシュメモリに保存するための第２の処理のプログラムコードとを書き込んだＲＯＭとを備えたことを特徴とする。

【００１２】また、本発明は、外部からソフトウェアをインストールして実行する機能を有する装置におけるソフトウェア・インストール方法であって、外部から入力された、ソフトウェアと、該ソフトウェアの正当性の検証のための、第１の鍵で暗号化されたシグネチャとを、バッファに一時的に保持するステップと、ＲＯＭに書き込まれた前記第１の鍵に一意に対応する第２の鍵と、前記バッファに一時的に保持されている前記ソフトウェアに対するシグネチャとに基づいて、前記ＲＯＭに書き込まれた第１の処理のプログラムコードによって、該ソフトウェアの正当性を検証するステップと、前記検証によって正当性が確認された場合に、前記ＲＯＭに書き込まれた第２の処理のプログラムコードによって、正当性が確認された前記ソフトウェアを前記バッファからフラッシュメモリに保存するステップとを有することを特徴とする。

【００１３】好ましくは、一定期間ごとに、プロセッサ及びワーキングメモリの初期化を行った後に、前記ＲＯＭに書き込まれた前記処理に制御を移すステップを更に有するようにしてもよい。

【００１４】なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

【００１５】本発明によれば、ＲＯＭに第２の鍵と検証のための第１の処理のプログラムコードと正当なソフトウェアをインストールするための第２の処理のプログラムコードとを作り込むとともに、入力したソフトウェア／シグネチャはバッファに一時的に保持した後に、ＲＯＭによって検証に成功したソフトウェアのみフラッシュメモリに保存することによって、鍵／検証処理／インストール処理の改竄を阻止し、正当性のあるソフトウェアのみをインストールすることを保証することが可能になる。また、これによって、正当性のあるソフトウェアの利用を保証することが可能になる。

【００１６】また、一定期間ごとにプロセッサ及びワーキングメモリの初期化を行った後に、検証／インストールの処理を行うことによって、動作中のソフトウェアが仮に改竄されたとしても、リセット後も改竄されたソフトウェアが継続して動作することを防ぐことができる。

【００１７】

【発明の実施の形態】以下、図面を参照しながら発明の実施の形態を説明する。

【００１８】図１に、本発明の一実施形態に係るシステムの全体構成例を示す。

【００１９】図１の機器１は、外部からインストールしたソフトウェアをプロセッサで実行する機能を有するものであれば、家電機器、汎用の計算機、携帯電話、ＰＤＡなど、どのような機器に対しても適用可能である。また、図１の機器１は、ＬＡＮ及び又はＷＡＮなどのネットワーク（無線ネットワークでも有線ネットワークでもよい）に接続可能な装置だけでなく、スタンドアローンの装置であってもよい。なお、以下では、主に機器１がネットワークに接続可能な家電機器である場合を例にとって説明している。

【００２０】インターネット３上に接続されるソフトウェア配布サーバ６は、「修正プログラム」、「修正プログラムを含んだアップデートプログラム」、「機能追加プログラム」など、当該機器１で動作するソフトウェアを公開し、配布する機能を持つ。ソフトウェア配布サーバ６は、例えば機器製造ベンダーもしくは機器に関連するソフトウェアの製造ベンダーが立ち上げたサーバでもよい。ソフトウェアには、当該ソフトウェアの配布元の秘密鍵で暗号化された、当該ソフトウェアに対するシグネチャが付加されているものとする。

【００２１】なお、ソフトウェア配布サーバ６は、必ずしもインターネット３上にある必要はなく、企業や個人の構築したＬＡＮ２上に設置してもよい。また、ＣＤ－ＲＯＭやフレキシブル・ディスクやＩＣカードなどの携帯可能な記録媒体によってソフトウェアを配布するようにしてもよい。また、インターネット３上にソフトウェア配布サーバ６を設ける方法と、ＬＡＮ２上にソフトウェア配布サーバ５を設置する方法と、記録媒体によってソフトウェアを配布する方法とのうちの２以上を組み合わせて実施してもよい。なお、以下では、主に機器１に既にインストールされているソフトウェアに対して上書きすべきソフトウェアを、配布・インストールする場合を例にとって説明している。

【００２２】なお、最初のバージョンのソフトウェアについては、上記と同じ方法で配布してもよいし、予め機器１にインストールされていてもよいし、その他の方法で配布・インストールされてもよい。

【００２３】機器１が外部からソフトウェアを取得する手段には種々の形態がある。例えば、（１）機器１が通信インタフェース部１５を備え、インターネット３上のソフトウェア配布サーバ６からソフトウェアをダウンロードする方法（この場合において、ＬＡＮ２を介してインターネット３に接続する方法、ＬＡＮ２を介さずにインターネット３に直接接続する方法などがある）、（２）機器１が通信インタフェース部１５を備え、ＬＡＮ２上のソフトウェア配布サーバ５からソフトウェアをダウンロードする方法（なお、この場合において、ソフトウェア配布サーバ５へは、例えばソフトウェア配布サーバ６等からインターネット３経由でソフトウェアを与える方法や、記録媒体からソフトウェアを読み込ませる方法などがある）、（３）機器１が外部記憶装置インタフェース部１６を備え、ＣＤ－ＲＯＭドライブやフレキシブル・ディスク・ドライブやＩＣカード・ドライブなどの外部記憶装置４からソフトウェアを読み込む方法などがある。

【００２４】もちろん、機器１は、上記に例示したような外部からソフトウェアを取得する手段を、１つのみ持っていてもよいし、複数持っていてもよい。

【００２５】ところで、ネットワークに接続された家電機器に対して、出荷後に追加的にソフトウェアをインストールし、そのソフトウェアを用いて利用者に対して新しいサービスを提供したり、ソフトウェアに欠陥が発見された場合に、家電機器製造ベンダやソフトウェアの製造ベンダがネットワーク経由で修正プログラムを配布し、インストールしたりすることは利用者、ベンダ双方にとって魅力的なサービスである。

【００２６】特に、家電機器は、利用者がネットワークセキュリティに対する知識を有しているとは限らず、さらに常にその機器のソフトウェアの状況を監視しているとは限らない。また、家電機器は、汎用の計算機に比べ物理的な被害をもたらす影響も大きいと考えられる。例えば、冷蔵庫の設計によっては、悪意のある利用者が冷蔵庫のソフトウェアを誤作動させることで、内容物を腐敗させる危険性がある。

【００２７】このため、利用者がセキュリティやソフトウェアに対する専門的な知識必要とせず、不正な作成者が作成したソフトウェアを排除して、正当なソフトウェアのみを適用させる仕組みを機器が備えるのが望ましい。

【００２８】図２に、ソフトウェアをインストールして実行する機能を有する機器１のハードウェア構成例を示す。

【００２９】図２に示されるように、機器１は、ＲＯＭ１１、フラッシュメモリ１２、ＲＡＭ１３、所定の処理を行うプロセッサ１４を備えている。

【００３０】ＲＯＭ１１には、ソフトウェア配布元の公開鍵と、ソフトウェアに付加されているシグネチャの検証処理（プログラム）と、ソフトウェアのインストール処理（プログラム）が作り込まれている。

【００３１】なお、当該機器１が複数のソフトウェア配布元を対象とする場合には、公開鍵は、ソフトウェア配布元に対応して複数備えればよい。なお、この場合には、公開鍵は、例えば、ソフトウェア配布元を識別するベンダーＩＤと対応付けてＲＯＭ１１に記憶し、ソフトウェアにはベンダーＩＤを付加して配布するものとすればよい。検証処理／インストール処理については、ソフトウェア配布元毎に用意する方法と、ソフトウェア配布元にかかわらずに同一にする方法とがある。

【００３２】また、当該機器１が複数のソフトウェア配布元を対象とする場合に、ＲＯＭ１１を、ソフトウェア配布元に対応してそれぞれ設ける構成も可能である。

【００３３】なお、検証処理プログラムとインストール処理プログラムとは、別々に作成されていてもよいし、一体化して作成されていてもよい。前者の場合には、検証処理が実行された後に、インストール処理が実行される形態や、最初にインストール処理が起動され、インストール処理が検証処理をコールする形態など、種々の形態が可能である。また、検証処理プログラムとインストール処理プログラムの他に、それら処理全体の制御を司る制御プログラムがさらにＲＯＭ１１に作り込まれている形態も可能である。

【００３４】ＲＡＭ１３は、プロセッサ１４が使用するワーキングメモリと、インストールする対象のプログラムを一時的に保存するバッファとを含む。この様子を図３に示す。なお、実際には、ワーキングメモリとバッファとは、同一のＲＡＭ上にあってもよいし、異なるＲＡＭ上にあってもよい（ワーキングメモリの一部とバッファの一部が同一のＲＡＭ上にあるなど、他の形態でもよい）。

【００３５】フラッシュメモリ１２は、ＲＡＭ１３のバッファに一時的に保存されているソフトウェアのうち、その正当性についての認証（シグネチャの検証）に成功したもののみを、記憶するためのものである。すなわち、ソフトウェア配布元により配布されるソフトウェアは、その認証を経て、フラッシュメモリ１２で保存する。

【００３６】ここで、図４に、ＲＯＭ１１とフラッシュメモリ１２とＲＡＭ１３との相互間での書き込み権限の関係を示す。フラッシュメモリ１２は、ＲＯＭ１１内に作り込まれたプログラムによってのみ、データを書き込むことができる。また、ＲＯＭ１１の内容は、改竄できないようになっている。したがって、シグネチャの検証を通過することのできない正当性のないソフトウェアが

フラッシュメモリ１２に保存されることはないので、フラッシュメモリ１２には、ソフトウェア配布元により配布された正当なソフトウェアが保存されている、ということが保証される。

【００３７】一方、機器1は、図2に示されるように、インストールするソフトウェアを入手するために、通信インターフェース部１５または外部記憶装置接続インターフェース部１６の少なくとも一方を備えている。なお、通信インターフェース部１５が利用するネットワークは、例えば、無線ネットワークでもよいし、イーサネット(登録商標)のような有線ネットワークでもよい。また、外部記憶装置接続インターフェース部１５は、ＣＤ－ＲＯＭなどの記録媒体を接続し、それら外部記憶装置とデータ交換可能な機能を備えるものである。

【００３８】なお、図2は、本実施形態を説明するために必要な部分のみ示したもので、実際には、図2に示された以外の種々の構成要素を備えて構わない。例えば、家電機器では、当該家電機器の本来の機能を実現するための構成要素が備わっている。

【００３９】図5に、機器1のソフトウェア構成例を示す。

【００４０】図5に示されるように、機器1は、オペレーティングシステム、アプリケーション、後述するタイマーを含む（必要に応じてミドルウェアが組み込まれる場合がある）。なお、タイマーは、ハードウェアで構成される場合もある。また、タイマーを備えない構成も可能である。

【００４１】なお、本実施形態における、機器へのインストール対象となるソフトウェアとは、機器を制御するオペレーティングシステム、ミドルウェア、及びそれら修正プログラム、機能追加プログラム、ユーザアプリケーションなどを指す。

【００４２】以下、図6を参照しながら、本実施形態についてより詳しく説明する。

【００４３】ここでは、主に、ソフトウェア配布サーバからネットワークを経由して家電機器1へソフトウェアをダウンロードし、シグネチャの検証を経た後に、ソフトウェアを上書きする場合を例にとって説明する。

【００４４】図7に、ソフトウェア取得時の手順の一例を示す。すなわち、機器1は、例えば予め登録された（1又は複数の）ソフトウェア配布サーバ（5または6）から必要なソフトウェアをダウンロードし（ステップＳ１）（図6の１０１参照）、ＲＡＭ１３のバッファ１３１に一時的に保存しておく（ステップＳ２）（図6の１０２参照）。なお、外部記憶装置インタフェース部１６による場合には、ＣＤ－ＲＯＭドライブにＣＤ－ＲＯＭを装着するなどして、ステップＳ１においてソフトウェアの読み込みを行う。

【００４５】サーバ登録については、（ｉ）機器出荷時に機器製造ベンダーまたはソフトウェアの製造ベンダーが規定してもよいし、（ii）出荷後に利用者が追加的に規定してもよいし、（iii）（i）と（ii）を両方可能としてもよい。ベンダーが規定したサーバ登録情報は、ＲＯＭ１１内に書き込まれていてもよい。

【００４６】ダウンロードすべきソフトウェアの有無のチェックについては、（ｉ）機器1に、定期的にソフトウェア配布サーバに対して、新しいアップデートするソフトウェアなどが配布（公開）されているかチェックする機能を持たせることで、利用者が意識することなく、自動的に最新のソフトウェアを機器1にインストールできるようにしてもよいし、（ii）利用者の要望に応じて、チェックする期間を変更したり、利用者が任意の時間にチェックをさせることができるようにしてもよいし、（iii）（i）と（ii）を両方可能としてもよい。

【００４７】ダウンロードにあたっては、機器1とソフトウェア配布サーバとの間で所定の認証／鍵交換手続きを行うことによって、セキュアな通信経路上でのデータ交換を行うようにしてもよい。

【００４８】なお、詳しくは後述する処理のために、ここでは、ダウンロード等したソフトウェアをＲＡＭ１３のバッファ１３１に保存するにあたっては、該ソフトウェアをＲＡＭ１３に保存した日時を該ソフトウェアに対応付けて保存しておくものとする。また、ソフトウェアをＲＡＭ１３のバッファ１３１からフラッシュメモリ１２に保存するにあたっては、該日時（該ソフトウェアをＲＡＭ１３に保存した日時）を該ソフトウェアに対応付けて保存しておくものとする。

【００４９】さて、ここで、機器1がソフトウェア配布サーバ（5または6）に接続して目的のソフトウェアをダウンロードする際、ネットワーク上の悪意ある利用者が、サーバのＩＰアドレスを詐称するなどして偽のサーバに誘導し、トロイの木馬やウィルスを含む偽のソフトウェアを機器1にダウンロードさせたり、ソフトウェア配布サーバ（5または6）に侵入したりしてサーバの内容を書き換えたりすることで偽のソフトウェアを配布する危険性がある。

【００５０】したがって、必ずしもソフトウェア配布サーバ（5または6）からダウンロードしたソフトウェアが正当であるとは限らないため、本実施形態では、ソフトウェアが悪意のある者によって作成された偽のソフトウェアではなく、正当なソフトウェアであるか検査し、正当なものと偽のものとを区別し、正当なもののみをインストールする機能を機器1に設けている。また、現在利用しているソフトウェアを改竄するなどによって上記の正当なもののみをインストールする機能自体を阻害する、ということができないようにしている。

【００５１】図8に、その検証処理／インストール処理の手順の一例を示す。

【００５２】前記のようにソフトウェアがダウンロードされＲＡＭ１３のバッファ１３１に一時的に保存される

一方で、図8の処理が行われる。

【0053】すなわち、タイマー（図6の17参照）が、一定時間の経過ごと（あるいは、一定の値をカウントするごと）に作動し、まず、プロセッサ14やRAM13のワーキングメモリ132のリセットを行う（ステップS11）（図6の103参照）。

【0054】このタイマーは、オペレーティングシステムやミドルウェア、アプリケーションの処理とは独立し、それらソフトウェアからは制御不可能で時間管理をする（例えば、ROM11に蓄えられた値を元に一定周期で値を加算し続ける）。一定時間が経過すると（あるいは、カウントが一定の値を越えると）、プロセッサ（のレジスタ等）と、一時的にワーキングメモリに蓄えられた現在のRAMの処理内容とを、無条件に強制的にクリアする。クリアとは、例えば初期化処理のような、それまでの処理状態を保存せずに値をリセットする処理を指す。

【0055】このタイマーは、値を加算し（もしくはメモリ範囲外の値まで加算された場合にはリセットし）、かつ減算することのできない機能を有する。該タイマーに対してソフトウェアがリセットをスキップすることで無効にしたり、タイマーの値を変更したりすることでリセットの時間を遅延させたりすることはできない。

【0056】なお、動作中のソフトウェアが強制的なリセットが原因でデータを損失・破壊してしまうことを防ぐために、ソフトウェアがタイマーに対してリセットを繰り上げる要求を発行してもよい。例えば、現在実行中のプログラムが次の処理単位を実行する前に、次のリセットまでの残り時間を調べ、もしもその時間が短かった場合には、（リセットに備えて）通常の処理手順は短時間でリセット可能な状態になるよう処理を行うようにしてもよい（この場合、次の処理単位は行わずに、短時間で済む処理を行うことによって、該処理がリセット時間前に完了した場合には、リセットを繰り上げて実行するようにすることも可能である）。

【0057】なお、機器1のコアの部分は、上記のリセットにかかわらずに、稼働し続けるものとしてもよい。例えば、インターネット機能を有する冷蔵庫において、インターネット機能の部分は、上記のリセットによって初期化されるが、冷蔵庫の本来の機能の部分は、上記のリセットにかかわらずに、稼働し続けるようにしてもよい。

【0058】さて、リセット処理を行った後、タイマーは、ROM11に処理を移す（図6の104参照）。

【0059】ROM11は、RAM12の特定のバッファ131をチェックする。

【0060】RAM13のバッファ131にソフトウェアのイメージが存在する場合（ステップS12）、そのソフトウェアのシグネチャと、ROM11に記憶された（該ソフトウェアの配布元の）公開鍵1101とを用いて、改竄検証処理（図6の105，106，1102参照）を行う（ステップS13）。

【0061】例えば、シグネチャが、当該ソフトウェアに所定のハッシュ関数を適用して得たハッシュ値を、該ソフトウェアの配布元の秘密鍵で暗号化したものである場合に、該シグネチャをROM11に記憶された公開鍵で復号して元のハッシュ値を求めるとともに、当該ソフトウェアに同一の所定のハッシュ関数を適用してハッシュ値を求め、両者が一致したならば、シグネチャの検証（ソフトウェアの認証）に成功したものとする。もちろん、検証処理については種々の方法が知られており、どのような方法を用いることも可能である。

【0062】ここで、改竄検証の結果が偽である場合（ステップS14）には、所定のエラー処理を行う（ステップS18）。

【0063】改竄検証に失敗する原因として、例えば、「ネットワークの異常によりファイルが正常に転送されなかった」、「ファイルが正当な作成者によって作成されたものではない」、などが考えられる。ファイルが正当な作成者によって作成されたものではない場合、悪意のある利用者が配布した偽のソフトウェアである可能性がある。そこで、例えば、エラー処理に、機器1からソフトウェアの配布元へエラーの情報をレポートする機能を備えれば、ソフトウェア配布元が対策や調査を行う上での資料として利用することが可能となる。

【0064】なお、検証に失敗したソフトウェアについては、エラー処理においてRAMのバッファから直ちに消去する方法、エラー処理においては消去しない（例えば、対策や調査等のための保存しておく）方法などがある。

【0065】他方、改竄検証の結果が真である場合（ステップS14）には、この手順例では、フラッシュ12に保存されている現在のソフトウェアに対応する日時（図6の107－1参照）と、RAM13のバッファ131上に記憶されたソフトウェアのイメージに対応する日時（図6の107－2参照）とを比較する処理（図6の1103参照）を行う（ステップS15）。

【0066】そして、RAM13のバッファ131上に記憶されたソフトウェアのイメージに対応する日時が、フラッシュ12に保存されている現在のソフトウェアに対応する日時より新しい場合（ステップS16）は、フラッシュメモリ12にRAM13のバッファ131のイメージをコピーし、以前の内容を上書きする（ステップS17）（図6の108参照）。

【0067】上記の比較処理の結果、RAM13のバッファ131上に記憶されたソフトウェアのイメージに対応する日時が、フラッシュ12に保存されている現在のソフトウェアに対応する日時より古い場合（ステップS16）は、既にそのソフトウェアがインストールされていると考えられるので（上記の検証に失敗したソフトウ

ェアをエラー処理においては消去しない場合には、検証に失敗したためにインストールしないことが決定されたソフトウェアであることもあり得る）、フラッシュ１２への書き込みは行わない。

【００６８】なお、ステップＳ１５においてフラッシュ１２に当該ソフトウェアが保存されていないことがあり得る構成の場合において、フラッシュ１２に当該ソフトウェアが保存されていなければ、ステップＳ１６では、例えば、ＲＡＭ１３のバッファ１３１上に記憶されたソフトウェアのイメージに対応する日時が、フラッシュ１２に保存されている現在のソフトウェアに対応する日時より新しいものとすればよい。また、ステップＳ１５においてフラッシュ１２に当該ソフトウェアが保存されていないことがあり得ない構成の場合において、フラッシュ１２に当該ソフトウェアが保存されていなければ、例えば、エラー処理を行えばよい。

【００６９】なお、ステップＳ１７において、フラッシュメモリ１２にＲＡＭ１３のバッファ１１３に記憶されたソフトウェアを書き込んだ際に、ＲＡＭ１３のバッファ１１３から当該ソフトウェアを消去するようにしてもよいし、その代わりにまたはそれとともに、ステップＳ１６においてＮｏの場合に、ＲＡＭ１３のバッファ１１３から当該ソフトウェアを消去するようにしてもよい。

【００７０】しかして、（ステップＳ１７でＲＡＭ１３のバッファ１１３からフラッシュメモリ１２にソフトウェアを書き込んだ場合、ステップＳ１６においてＮｏの場合、ステップＳ１８でエラー処理を行った場合、またはステップＳ１２においてＮｏの場合（ＲＡＭ１３のバッファ１３１にソフトウェアのイメージがない場合）に、）ＲＯＭ１１は、フラッシュメモリ１２に処理を移動させ、フラッシュメモリ１２に書き込まれたイメージに制御を移し、当該イメージはシステムの起動処理（図６の１０９参照）を実行する。

【００７１】ここで、本実施形態のメモリアドレス制御機能について説明する。

【００７２】かりにＲＡＭ１３のワーキングメモリ１３２上で動作するソフトウェアが改竄されてしまったとする。しかし、上記のように、タイマーによるリセット時にＲＡＭ１３のワーキングメモリ１３２はリセットされる。従って、この改竄されたソフトウェアがリセット後も存続するには、フラッシュメモリ１２にコピーする必要がある。しかしながら、フラッシュメモリ１２については、ＲＯＭ１１からの書き換えのみを可能とするため（図４参照）、改竄されたＲＡＭ１３のイメージがフラッシュ１２を書き換えることはできない。このメモリアドレス制御手段を用いることで、改竄されたＲＡＭ１３のイメージがリセット後も存在し続けることはない。

【００７３】タイマーによる強制的なリセット手段と、メモリアドレス制御手段により、ＲＡＭ１３のワーキングメモリ１３２で動作中のソフトウェアがかりに悪意のある利用者によって改竄されてしまった場合でも、次回の起動時にフラッシュメモリ１２に保存された正当なソフトウェアに復旧することが可能となる。

【００７４】以上のように、本実施形態によれば、インストール対象のソフトウェアを、正当な作成者によって作成されたことが保証されたものと、それ以外のもの（例えば、偽造されたもの）とに区別し、前者のみを選別しインストールすることが可能となる。また、かりにソフトウェアが改竄されたとしても、定期的にシステムをリセットすることによって、現在のワーキングメモリに存在する改竄されたイメージが存在し続けることを阻止できる。従って、機器は常に正当な作成者が作成したソフトウェアを利用することが保証され、かつ正当な作成者が作成したソフトウェアのみをインストールすることができる。

【００７５】なお、図８の手順は一例であり、種々変形して実施することが可能である。例えば、上記処理では、まず、シグネチャによる検証処理を行い、これに成功した場合に、ＲＡＭのバッファ上に保存されたソフトウェアとフラッシュメモリに保存されたソフトウェアとの日時を比較する日時チェックを行い、さらにこれに成功したならば、ダウンロードされたソフトウェアが正当かつインストールすべきものであると判断して、フラッシュメモリにＲＡＭのバッファのイメージを書き込んだが、逆に、日時チェックを行い、これに成功した場合に、検証処理を行い、さらにこれに成功したならば、フラッシュメモリに書き込むようにしてもよい。また、検証処理と日時チェックを並列的に行い、それらの結果が両方とも真である場合に、フラッシュメモリに書き込むようにし、少なくとも一方の結果が偽であれば、書き込まないようにしてもよい。

【００７６】また、これまでの説明では、同一性を有するソフトウェアに対して「上書きするソフトウェア」を想定しているが、メインとなるソフトウェアに対してソフトウェアを追加する場合には、例えば図８の手順において、ステップＳ１５とステップＳ１６を省き、ステップＳ１７の上書き処理を組み込み処理に修正して、ステップＳ１４で検証に成功した場合にステップＳ１７で該当ソフトウェアを組み込むようにすればよい。また、その時々で上書きするか追加するかが異なってくる場合には、例えば図８の手順においては、対象としているソフトウェアを上書きすべきか追加すべきかを判断し、前者の場合には、日時チェックを行い、（検証処理に）日時チェックの結果が真のときに、フラッシュメモリにＲＡＭのバッファのソフトウェアを上書きし、後者の場合には、日時チェックを行わずに、フラッシュメモリのソフトウェアにＲＡＭのバッファのソフトウェアを追加するようにしてもよい。なお、対象としているソフトウェアを上書きすべきか追加すべきかについては、例えば、各ソフトウェアに固有の識別情報を持たせ、ＲＡＭのバッ

ファ内のソフトウェアの持つ識別情報と同一の識別情報を持つものが、フラッシュメモリに保存されていれば、上書きすると判断し、保存されていなければ、追加すると判断するようにしてもよい。また、例えば、各ソフトウェアに、上書きすべきか追加すべきかを示す情報を付加する方法もある。

【００７７】また、図８のステップＳ１２で複数種類のソフトウェア（各ソフトウェアは、例えば、ソフトウェアに固有の識別情報によって識別される）がＲＡＭのバッファに存在し得る場合には、検証／インストールを、ＲＡＭのバッファに存在する各々のソフトウェアに対して行えばよい。

【００７８】また、図８のステップＳ１９のフラッシュメモリ内のソフトウェアの起動にあたって、起動できるソフトウェアがフラッシュメモリ内に複数存在し得る場合には、予め定められた選択基準に従って、起動すべきものを選択して、起動すればよい。この選択基準としては、例えば、予めＲＯＭに設定された識別情報を持つソフトウェアを選択する方法、予めユーザが指定したソフトウェアを選択する方法、リセット時に起動中であったものを選択する方法など、種々の方法がある。

【００７９】また、図８の手順においては、ステップＳ１９のフラッシュメモリ内のソフトウェアの起動を行わず、異なる契機によって起動を行う方法もある。

【００８０】また、図８の手順においては、ステップＳ１１のプロセッサやＲＡＭのワーキングメモリのリセットを行わず、異なる契機によって行う方法もある。

【００８１】また、これまでタイマーの起動時にソフトウェアが自動的にインストールされるような方法を述べてきたが、自動インストールの代わりにまたはこれとともに、利用者が任意の時間にソフトウェアをインストールできるようにすることも可能である。このようにすることによって、例えば、ソフトウェアの重要な欠陥が発見された場合に、利用者の希望に応じて修正プログラムをインストールすることができる。

【００８２】利用者が任意の時間にインストールする機能については、自動インストールと併用する場合には、例えば、前述したタイマーのリセットを繰り上げて実行する機能があれば、この機能を利用して実現することもできる。タイマーのリセットを繰り上げる命令を実行するにあたっては、機器１に物理的なスイッチを備えてもよいし、赤外線リモコンなどを通じて機器１に要求を送信するようにしてもよい。また、ネットワーク経由で任意のコマンドを送信することも可能である。この場合、タイマーがリセット要求を受けると、図８の手順と同様に、プロセッサの現在の処理内容とＲＡＭのワーキングメモリに保存された情報をリセットし、ＲＯＭに処理内容を移し、ソフトウェアの改竄検証及びインストールを開始する。

【００８３】自動インストール機能を備えずに、利用者が任意の時間にソフトウェアをインストールする機能を備える場合には、ユーザの入力を契機として、リセット、検証、インストールの処理を行えばよい（この場合に、リセットは行わない形態等も可能である）。

【００８４】また、これまでの説明では、ダウンロード等したソフトウェアをＲＡＭのバッファに保存するにあたって、該ソフトウェアをＲＡＭに保存した日時を該ソフトウェアに対応付けて保存しておき、ソフトウェアをＲＡＭのバッファからフラッシュメモリに保存するにあたっては、該日時（該ソフトウェアをＲＡＭに保存した日時）を該ソフトウェアに対応付けて保存しておき、両者の日時を比較した結果に応じて、ソフトウェアをＲＡＭのバッファからフラッシュメモリに保存するか否かを決定したが、その代わりに、例えば、ソフトウェアにバージョン情報等を付加しておき、両バージョン情報の新旧の比較結果に応じて、ソフトウェアをＲＡＭのバッファからフラッシュメモリに保存するか否かを決定するようにしてもよい。また、その他の方法も可能である。

【００８５】また、これまでは、ソフトウェアには、その配布元の秘密鍵で暗号化されたシグネチャが付加され、機器１のＲＯＭ１１には該ソフトウェア配布元の公開鍵が作り込まれているものとして説明したが、公開鍵暗号方式ではなく、他の暗号方式を採用することも可能である。例えば、ソフトウェア配布元と機器１とで同一の共有鍵を用いる方法を採用することも可能である。

【００８６】また、これまでネットワークを経由したソフトウェアの配布とインストールを中心に説明してきたが、前述したように、機器または利用者がネットワーク経由でソフトウェアを入手する必要があるわけではない。機器が通信インターフェースを持っていない場合や、機器がネットワークに接続されていない場合は、フロッピー(登録商標)ディスクなどの携帯可能で書き込み可能なメディアや、ＣＤ－ＲＯＭなどのような携帯可能で書き込み不可能なメディアからソフトウェアをアップデートすることが考えられる。もちろん、ネットワーク経由でのダウンロードの場合と同様に、記憶媒体となるメディアは必ずしも正当な作成者が作成したソフトウェアが記録されているという保証はないため、一旦ＲＡＭのバッファに保存した後にフラッシュメモリにインストールすることで適用可能である。

【００８７】また、前述したように、本発明は、家電機器に限らず、汎用の計算機や、携帯電話、ＰＤＡなどの機器に対しても適用可能である。特に、携帯電話やＰＤＡは、メモリの容量に制限があるため、一度に大量のアプリケーションをインストールすることが困難である。また、携帯電話のサービスやインターネット接続サービスを利用してインターネット経由でアプリケーションをダウンロードし、インストールする際に、本発明が適用可能である。

【００８８】なお、この発明の実施の形態で例示した構

成は一例であって、それ以外の構成を排除する趣旨のものではなく、例示した構成の一部を他のもので置き換えたり、例示した構成の一部を省いたり、例示した構成に別の機能あるいは要素を付加したり、それらを組み合わせたりすることなどによって得られる別の構成も可能である。また、例示した構成と論理的に等価な別の構成、例示した構成と論理的に等価な部分を含む別の構成、例示した構成の要部と論理的に等価な別の構成なども可能である。また、例示した構成と同一もしくは類似の目的を達成する別の構成、例示した構成と同一もしくは類似の効果を奏する別の構成なども可能である。また、この発明の実施の形態で例示した各種構成部分についての各種バリエーションは、適宜組み合わせて実施することが可能である。また、この発明の実施の形態は、個別装置としての発明、関連を持つ2以上の装置についての発明、システム全体としての発明、個別装置内部の構成部分についての発明、またはそれらに対応する方法の発明等、種々の観点、段階、概念またはカテゴリに係る発明を包含・内在するものである。従って、この発明の実施の形態に開示した内容からは、例示した構成に限定されることなく発明を抽出することができるものである。

【００８９】本発明は、上述した実施の形態に限定されるものではなく、その技術的範囲において種々変形して実施することができる。

【００９０】

【発明の効果】本発明によれば、正当でないソフトウェアのインストールを阻止することができる。

【図面の簡単な説明】

【図１】本発明の一実施形態に係るシステムの全体構成例を示す図

【図２】同実施形態に係る機器のハードウェア構成例を示す図

【図３】同実施形態に係る機器のＲＡＭの構成例を示す図

【図４】同実施形態に係る機器のＲＯＭとフラッシュメモリとＲＡＭとの相互間での書き込み権限の関係を示す図

【図５】同実施形態に係る機器のソフトウェア構成例を示す図

【図６】同実施形態に係る機器内でのデータの流れ及び制御について説明するための図

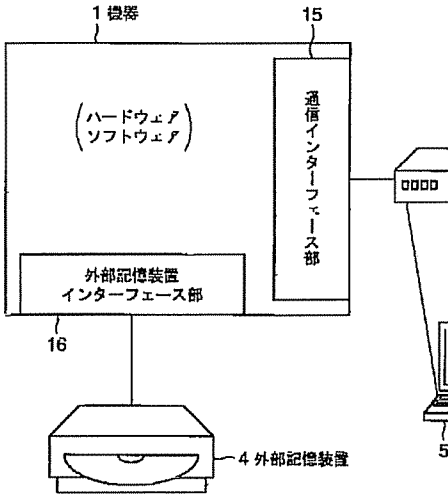
【図７】同実施形態に係る機器のソフトウェア取得時の手順の一例を示すフローチャート

【図８】同実施形態に係る機器の認証処理／インストール処理の手順の一例を示すフローチャート

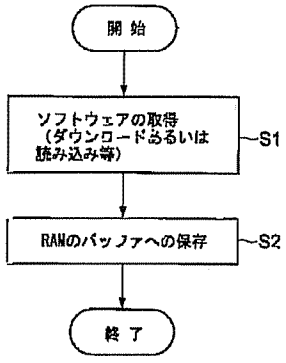
【符号の説明】

１…機器
２…ＬＡＮ
３…インターネット
４…外部記憶装置
５，６…ソフトウェア配布サーバ
１１…ＲＯＭ
１２…フラッシュメモリ
１３…ＲＡＭ
１４…プロセッサ
１５…通信インターフェース部
１６…外部記憶装置接続インターフェース部
１７…タイマー

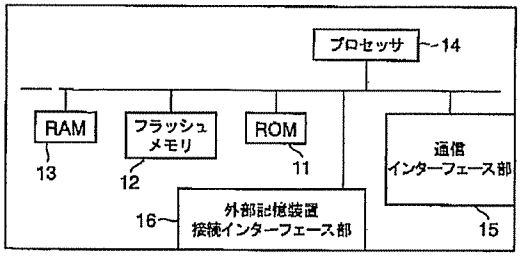
【図１】



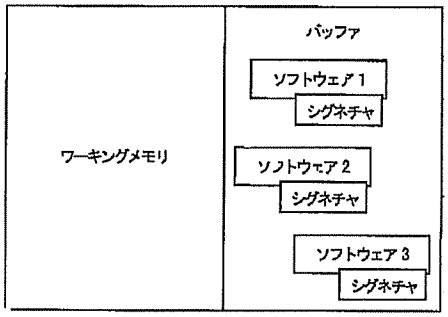
【図７】



【図2】



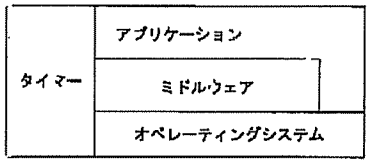
【図3】



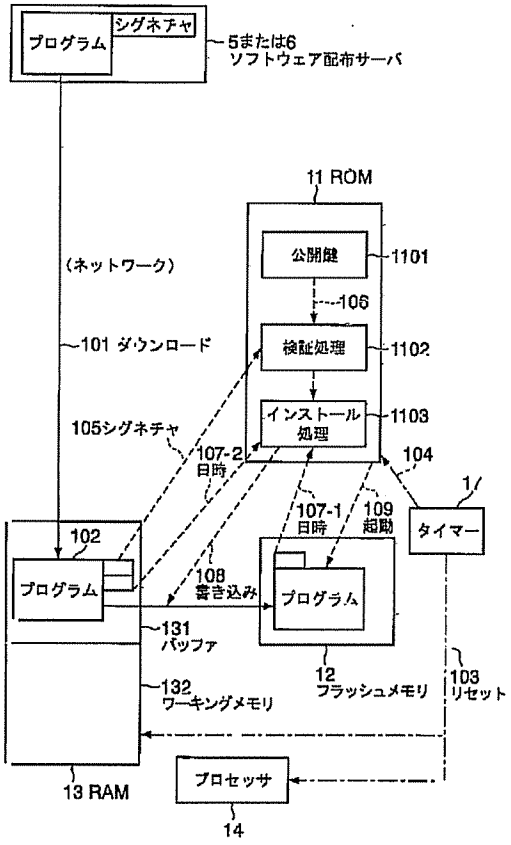
【図4】

書き込み先	書き込み元			
		ROM	フラッシュメモリ	RAM
	ROM	・・・・	・・・・	・・・・
	フラッシュメモリ	○	×	×
	RAM	○	○	○

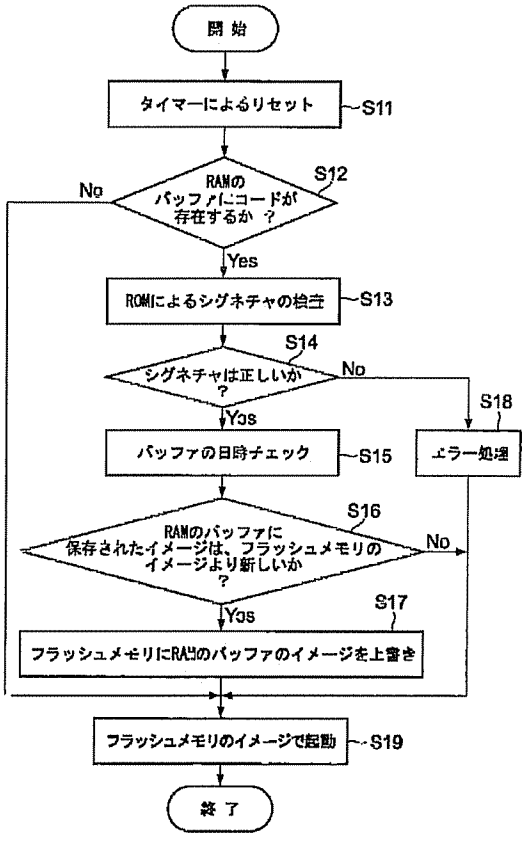
【図5】



【図6】



【図8】



フロントページの続き

Fターム(参考) 5B017 AA02 BA02 BA07 BA09 CA15
5B076 AC01 AC05 AC10 BA05 BB04
BB06 BB13 FA13 FA20 FB11